# MapReduce mérés

A mérést kidolgozta: Prof. Do Van Tien, Dr. Do Hoai Nam

## Bevezetés

A mérés célja a MapReduce paradigma ismertetése Apache Hadoop keretrendszerben (Apache Hadoop Yarn, HDFS). A mérés során megismerkedünk a MapReduce folyamattal, valamint a HDFS-nek MapReduce-ra való hatásaival az applikációk végrehajtása során.

A mérés kiértékeléséhez jegyzőkönyvet kell készíteni, amelyet a https://www.office.com/-ra (BME account) kell feltölteni. Utána meg kell adni a hozzáférést do.hoainam@vik.bme.hu felhasználónak.

**Segédanyagok a felkészüléshez**:

- HDFS: https://www.edureka.co/blog/apache-hadoop-hdfs-architecture/
- Apache Hadoop YARN: https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YARN.html
- MapReduce: https://www.edureka.co/blog/mapreduce-tutorial/
- EnforcementDesign cikk: http://www.hit.bme.hu/~do/papers/EnforcementDesign.pdf

## Eszközök és programok

Az Apache Hadoop keretrendszer (Yarn, HDFS) egy VM gépen van telepítve. A VM gép hozzáféréséhez az **x2goclient** programot használjuk távolról.

# Mérési feladatok

**Előkészítés**

Indítsuk el az Apache Yarn klasztert és a HDFS-t. Egy terminálban:

*start-dfs.sh*

*start-yarn.sh*

*mapred --daemon start historyserver*

A **jps** paranccsal tudjuk ellenőrizni, hogy minden szükséges Hadoop komponens sikeresen elindult-e.

A HDFS GUI a http://localhost:9870 linken érhető el. Az Apache Yarn GUI http://localhost:8088 linken érhető el.

**Első feladat**

Ebben a feladatban megismerkedünk a MapReduce folyamattal. A MapReduce és a HDFS rövid ismertetője a https://www.edureka.co/blog/mapreduce-tutorial/ és https://www.edureka.co/blog/apache-hadoop-hdfs-architecture/ linkeken található meg.

Lépések:

- Másoljuk át a */home/meres/task1* mappában levő txt fájlokat HDFS */home/meres/task1* mappába

  *hdfs dfs -mkdir task1*

  *hdfs dfs -put /home/meres/task1/*.txt task1*

- Futtassuk a MapReduce WordCount applikációt

  *hadoop jar $MR_APP_JAR wordcount -Dmapreduce.job.queuename=tq01 task1 task1out*

- Eredmény megtekintése:

  *hadoop fs -cat task1out/part-r-00000*

Elemzés:

- Hány fájl található a HDFS */home/meres/task1* könyvtárban? Hány HDFS block van az egyes fájlokban?

- Hány Map task és Reduce task keletkezett az applikáció futtatása során? Mi az összefüggés a HDFS */home/meres/task1* könyvtárban levő fájlok száma és a Map taskok száma között?

- Hasonlítsuk össze az applikáció eredményét a */home/meres/task1*/*results.txt* fájlban levő eredménnyel!

**Második feladat**

Futtassuk újra MapReduce WordCount applikációt nagy méretű fájlokra

*hadoop jar $MR_APP_JAR wordcount -Dmapreduce.job.queuename=tq01 task2 task2out*

Elemzés (a jegyzőkönyvhöz):

- Hány fájl és HDFS blokk van összesen a HDFS /home/meres/task2 mappában?

- Hány Map task és Reduce task keletkezett az applikáció futtatása során? Mi az összefüggés a HDFS blokkok száma és a Map taskok száma között?

**Harmadik feladat**

Ebben az feladatban megnézzük mi történik, ha több Mapreduce task olvas egyszerre adatot a HDFS-ről.

*Első eset*:

- Network throughput monitorozása

  *net-capture-start.sh* ~/captures/task3a *TestDFSIO*

- TestDFSIO teszt futtatása egy külön terminálban

  *hadoop jar $MR_TEST_JAR TestDFSIO -Dmapreduce.job.queuename=tq03 -read -nrFiles 1 -fileSize 2048*

- Monitorozás leállítása az applikáció befejezése után és a diagramok generálása

  *net-capture-stop.sh*

  *result-process.sh ~/captures/task3a TestDFSIO*

*Második eset*:

- Network throughput monitorozása

  *net-capture-start.sh* ~/captures/task3b *wordcount*

- Wordcount futtatása egy külön terminálban

  *hadoop jar $MR_APP_JAR wordcount -Dmapreduce.job.queuename=tq01 largefiles task3bout*

- Monitorozás leállítása az applikáció befejezése után és a diagramok generálása

  *net-capture-stop.sh*

  *result-process.sh ~/captures/task3b wordcount*

*Harmadik eset*:

- Network throughput monitorozása

  *net-capture-start2.sh* ~/captures/task3c *TestDFSIO wordcount*

- TestDFSIO és Wordcount párhuzamos futtatása

  *hadoop jar $MR_APP_JAR wordcount -Dmapreduce.job.queuename=tq01 largefiles task3cout*

  *(Néhány másodperccel később másik terminálban ):*

  *hadoop jar $MR_TEST_JAR TestDFSIO -Dmapreduce.job.queuename=tq03 -read -nrFiles 1 -fileSize 2048*

- Monitorozás leállítása az applikációk befejezése után és a diagramok generálása

  *net-capture-stop.sh*

  *result-process2.sh ~/captures/task3c TestDFSIO wordcount*

Elemzés:

- Mennyi a read sebesség a TestDFSIO applikáció eredménye alapján egyes esetekben?

- Mennyi a TestDFSIO és WordCount futtatási ideje egyes esetekben?

- Generált diagramok összehasonlítása

**<u>Negyedik feladat</u>**

Az előző feladatban kapott diagramok alapján látni lehet a taskok közötti *versengést* a diszk I/*O*-ért.

A Map taskok HDFS olvasási sebesség korlátozására alkalmazzuk a http://www.hit.bme.hu/~do/papers/EnforcementDesign.pdf cikkben javasolt megoldást.

Az alábbi három esetben a TestDFSIO/WordCount HDFS (per task) maximális olvasási sebességét korlátozzuk a következő értékekre

- A eset: 30/30 mbps,
- B eset: 10/30 mbps
- C eset: 50/30 mbps.

*Lépések az első eset elemzéséhez*:

- Network throughput monitorozása

  *net-capture-start2.sh* ~/captures/**task4a** *TestDFSIO wordcount*

- Map taskok HDFS olvasási sebességének korlátozása

  *net-control-start2.sh TestDFSIO 30 wordcount 30*

- TestDFSIO és Wordcount párhuzamos futtatása

  *hadoop jar $MR_APP_JAR wordcount -Dmapreduce.job.queuename=tq01 largefiles task4aout*

  *(Néhány másodperccel később másik terminálban ):*

  *hadoop jar $MR_TEST_JAR TestDFSIO -Dmapreduce.job.queuename=tq03 -read -nrFiles 1 -fileSize 2048*

- Monitorozás leállítása az applikációk befejezése után és a diagramok generálása

  *net-control-stop.sh*

  *net-capture-stop.sh*

  *result-process2.sh ~/captures/t**ask4a** TestDFSIO wordcount*

  Elemzés:

  - Mennyi a read sebesség a TestDFSIO applikáció eredménye alapján az egyes esetekben?
  - Mennyi az applikációk futtatási ideje az egyes esetekben?
  - A generált diagramok összehasonlítása!

Ezeket a lépéseket ismételjük a második és harmadik esetekre!